Homogenized Yarn-Level Cloth

GEORG SPERL, IST Austria
RAHUL NARAIN, Indian Institute of Technology Delhi
CHRIS WOJTAN, IST Austria

Fig. 1. Left: A comparison between direct yarn-level simulation (YLC) and simulation with our homogenized model (HYLC); our homogenized model accurately captures the non-trivial elastic stretching and bending response of the fabric. Middle and right: Results simulated with homogenized continuum models of woven and knitted patterns; our method allows us to efficiently compute large-scale simulations where direct yarn-level simulation would be prohibitively slow.

We present a method for animating yarn-level cloth effects using a thin-shell solver. We accomplish this through numerical homogenization: we first use a large number of yarn-level simulations to build a model of the potential energy density of the cloth, and then use this energy density function to compute forces in a thin shell simulator. We model several yarn-based materials, including both woven and knitted fabrics. Our model faithfully reproduces expected effects like the stiffness of woven fabrics, and the highly deformable nature and anisotropy of knitted fabrics. Our approach does not require any real-world experiments nor measurements; because the method is based entirely on simulations, it can generate entirely new material models quickly, without the need for testing apparatuses or human intervention. We provide data-driven models of several woven and knitted fabrics, which can be used for efficient simulation with an off-the-shelf cloth solver.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: knitted, woven, cloth simulation, yarn-level cloth, homogenization, data fitting

ACM Reference Format:
Georg Sperl, Rahul Narain, and Chris Wojtan. 2020. Homogenized Yarn-Level Cloth. *ACM Trans. Graph.* 39, 4, Article 31 (July 2020), 14 pages.

1 INTRODUCTION

The simulation and analysis of yarn-level cloth has recently generated a great deal of research in the computer graphics [????], materials science [??], and physics communities [?]. Woven and knitted materials can exhibit a wide array of behaviors (highly variable stretchiness, anisotropy, area-preservation effects, etc.). They can be produced from simple threads or wires, leading to their ubiquity in everyday life. Furthermore, these materials are fascinating from a theoretical point of view, because their varying material properties arise almost entirely from the *geometric structure* of the threads — subtly different stitch patterns can lead to dramatically different material behaviors. For example, Figure 2 illustrates how different knit patterns influence area-preservation and curling of the fabric.

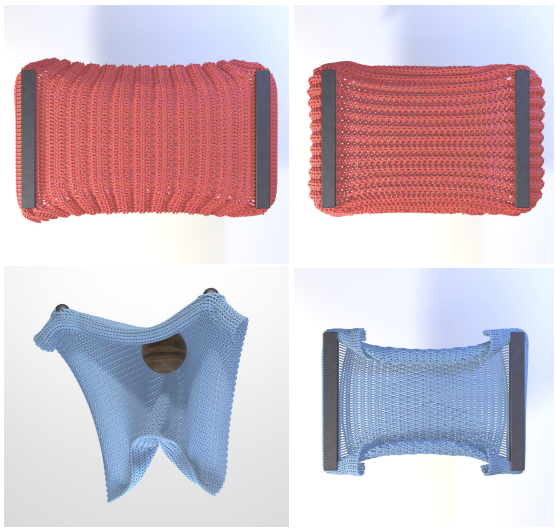


Fig. 2. Large-scale phenomena can emerge from simulated yarn-level geometry. The rib pattern (top) exhibits anisotropy and a tendency to preserve area under tension, while the stockinette pattern (bottom) exhibits curling.

Authors' addresses: Georg Sperl, IST Austria, georg.sperl@ist.ac.at; Rahul Narain, Indian Institute of Technology Delhi, narain@cse.iitd.ac.in; Chris Wojtan, IST Austria, wojtan@ist.ac.at.

2020. 0730-0301/2020/7-ART31
https://doi.org/

Simulating woven and knitted materials as a collection of interacting threads can accurately reproduce highly complex behaviors, but this direct strategy tends to be computationally expensive. On the other hand, finite element-based cloth simulations are relatively computationally efficient, because they replace the simulation of individual fibers with an approximate material model based on continuum mechanics. Unfortunately, choosing a suitable material model is a nontrivial task, and Little is known about the continuum behavior of many woven and knitted fabrics in particular, so many yarn-level effects cannot be captured by existing material models in computer graphics.

In this work, we aim to determine material properties directly from yarn-level geometry using numerical homogenization. We precompute the effective material response from periodic yarn-level simulations, learn an approximate material model from the resulting data, and incorporate this new material into an off-the-shelf cloth simulator. We offer the following technical contributions:

- The first use of numerical homogenization for animating woven and knitted fabrics
- Novel co-rotated periodic boundary conditions for the nonlinear homogenization of thin shells
- A procedure for fitting a material model capable of reproducing common textile phenomena such as anisotropy, area-preservation, and curling.

2 RELATED WORK

This paper incorporates ideas from a variety of topics including the simulation of rods and shells, data-driven models, and homogenization. In the following, we only provide a brief summary of closely related areas.

2.1 Cloth simulation

Yarn-Level Cloth. Modern simulators approximate the behavior of an individual strand of yarn or thread using the theory of elastic rods [???]. Simulation of fabric at the yarn level was pioneered in computer graphics by **?** with subsequent work on improving the treatment of collision handling [?] and using persistent contacts to simulate woven [?] and knitted fabrics [??]. **?** propose a method for the interactive authoring and editing of small periodic yarn patches on GPUs, and they reproduce the rest shape of a multitude of stitches and patterns under tension. **?** introduce stitch meshes to create large-scale virtual knitted patterns, and **?** ensure they are actually fabricable. **??** introduce a suite of tools for converting virtual knit patterns into garments fabricated by a machine. MPM modeling of yarn-level simulations was first performed by **?**, and then combined with neural networks for homogenizing fiber-level motions to yarn level motions [?]. In our work, we rely on the methods of **?** and **?** for the simulation of our periodic yarn patterns.

Continuum-Level Cloth. Researchers in computer graphics often treat cloth as an elastic solid with a potential energy that increases as it deforms from its rest state. Typical methods for discretizing such an elastic solid are mass-spring networks [???], discrete thin shells [?], and continuum mechanics solvers based on finite differences [?], finite elements [??], and the material point method [?]. We use a finite-element thin-shell solver to simulate our macroscale cloth (ArcSim [??]).

Data-driven Cloth. Many of the methods above use analytically derived material models based on a somewhat straightforward relationship between deformation and potential energy. However, the material model can also be learned from example data. **?** propose an inexpensive setup for measuring features of fabric under tensile and bending tests, and they optimize piece-wise linear material models. **?** develop a measurement setup to capture more complex 3D deformations of cloth with complete position and force data. **?** show how to inexpensively measure internal stretching and bending friction, and they optimize the required parameters based on sparse data. Further research discusses incremental fitting of separable models for convex hyperelastic materials [?] and an orthotropic model for woven fabric based on commercially available tests [?].

For each update to model parameters, the above methods typically need to recompute quasistatic cloth equilibria to compare to real-world measurements. They also mention difficulty in accurately capturing bending. In our approach, data-gathering and fitting are decoupled. We precompute deformation responses once as an inexpensive preprocessing step and thus do not require simulations during fitting. We also do not require any real-world measurement setup. Additionally, our method can directly compute the bending resistance for applied curvatures, allowing for more controlled measurements.

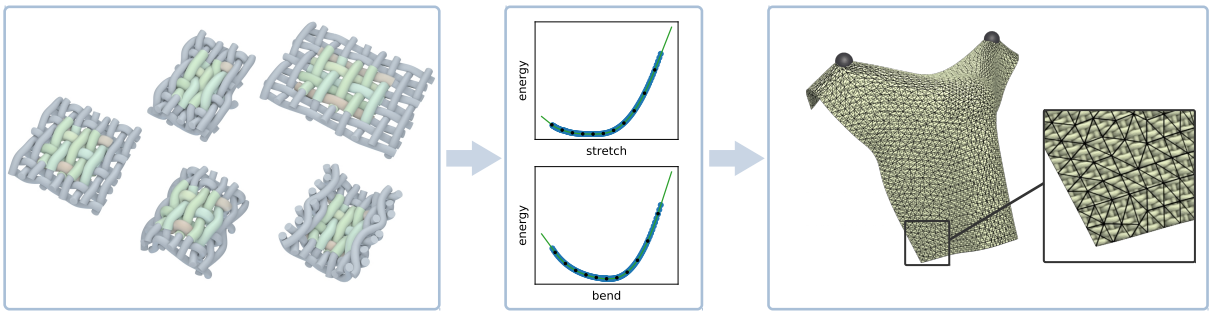
Finally, data-driven methods have also been used to add detail to coarse simulations; **?** add detailed wrinkles to coarse simulations at interactive rates using a database of precomputed high-resolution simulations.

2.2 Multiscale modeling and homogenization

In computer graphics, the concept of multiscale modeling covers a wide area of research such as analytic multiscale models [???], numerical coarsening [???], meta-materials and digital fabrication [????], sound simulation [?], and rendering [??].

Our work focuses on homogenization of periodic yarn patterns and is thus closely related to the work of **?**. They investigate the elastic properties of isohedral tilings represented as planar rod patterns through numerical homogenization. They also provide a tool for exploring the various families of tilings and discuss emergent properties such as material symmetries in detail. Their tool examines material nonlinearities by fitting linear models at multiple magnitudes of deformation. Our work can be seen as an extension to fully non-linear models for non-planar woven and knitted yarn patterns. In addition, our novel boundary conditions let us homogenize interaction between multiple modes of deformation, such as simultaneous stretching and bending.

Computational Homogenization. Multiscale modeling has received a lot of attention also outside of computer graphics; this includes the technique of computational homogenization, where macroscopic material responses are computed based on representative microscale simulations [??]. Macroscopic strains are imposed on the representative microscale material sample through boundary conditions, and stresses can be computed through averaging. To this end, **?** and **?**



(a) Precomputed Homogenization (b) Fitting (c) Thin-shell Simulation

Fig. 3. Our method takes a periodic yarn pattern and produces a homogenized cloth material model. (a) We impose macroscopic in-plane and bending deformations on a periodic pattern. (b) We compute homogenized energy density samples for ranges of deformations and fit them with regularized splines. (c) The resulting material model can be used to efficiently simulate cloth by computing elastic responses of the pattern to deformations.

propose a generalized framework to derive microscale boundary conditions and averaging relations for homogenization in general. For more details, we refer to the reviews of **?** and **?**.

This method has been applied to the homogenization of thin shells [**?**] as well as textiles and fabrics [**??**]. However, they use a small-curvature assumption which is inadequate for large bending, as we will discuss in Section 4.2.

The nature of representative microscale computations in computational homogenization lends itself to data-driven approaches. Various approaches fit constitutive models from precomputed stress and energy data [**???**]. However, the basic constitutive models used are either not descriptive enough for our data or do not provide any guarantees to ensure smooth animation.

Other Continuum Models for Fabric. The physics and engineering communities have also developed continuum-level models for approximating the behavior of fabrics. **?** and **?** propose mathematical models describing the rich material response of a stockinette pattern based on inextensible and incompressible yarns. However, their investigations are limited to a small set of extension tests. Researchers have also developed mesoscopic models of woven fabric using spring-based finite elements [**???**].

3 OVERVIEW

We first explain the necessary background in homogenization and derive novel periodic boundary conditions for our method in Section 4. We then implement these ideas in a quasi-static yarn-level cloth simulator, as described in Section 5. Using this microscale yarn-level simulator, we sample the material's behavior in response to a number of different in-plane and bending deformations. We use regularized spline regression to fit a macroscopic energy density model to this data (Section 6), and use the new material model directly in a thin-shell cloth simulator (Section 7). Figure 3 provides an overview of our method.

Modeling Assumptions. Woven and knitted fabrics are complex materials with non-trivial elastic, plastic, hysteretic, and damping behaviors. As a first step toward data-driven yarn-level cloth simulation, this paper assumes that these materials exhibit a purely hyperelastic response to deformation. Although our current approach is limited, we show in Section 8 that this hyperelastic assumption is sufficient to reproduce a number of qualitative effects specific to yarn-level materials. We discuss future extensions in the directions of data-driven plasticity, hysteresis, and damping in Section 9.

4 HOMOGENIZATION

We begin by summarizing the "kinematic averaging" theory of computational homogenization for volumetric solids, and we extend these concepts to the homogenization of thin shells in the second part of this section. For further details, we recommend the following reviews on computational homogenization and multiscale modeling [**??**].

We use the terms *microscopic* and *microscale* when referring to small local (yarn-level) effects, and we use the terms *macroscopic* and *macroscale* when referring to average (continuum-level) behaviors of the bulk material. We write macroscopic quantities $\overline{\boldsymbol{x}}$ with a bar and microscopic quantities $\boldsymbol{x}$ without. We use Latin indices i, j to iterate dimensions 1, 2, 3, and Greek indices α, β to iterate only the first two dimensions 1, 2. We use indices preceded by a comma as shorthand for derivatives, e.g. $x_{i,j}$ is the derivative of element x_i with respect to parameter j.

4.1 Computational Homogenization of Volumetric Solids

We describe the macroscale deformation of an elastic solid with reference coordinates $\overline{\boldsymbol{X}}$, deformed coordinates $\overline{\boldsymbol{x}}$, and deformation gradient $\overline{\boldsymbol{F}} = \frac{\partial \overline{\boldsymbol{x}}}{\partial \overline{\boldsymbol{X}}}$. Similarly, we have microscale quantities $\boldsymbol{X}$, $\boldsymbol{x}$, and $\boldsymbol{F} = \nabla \boldsymbol{x} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}$. Homogenization theory assumes that the bulk material exhibits microscale variations, and thus we can zoom in at any macroscale point $\overline{\boldsymbol{x}}$ to find a volume of microscale material, called the *representative volume element* (RVE) [**??**]. Mathematically, we can describe the RVE with a first-order expansion about a point $\overline{\boldsymbol{x}}$ [**?**]:

$$\boldsymbol{x}(\boldsymbol{X}) = \overline{\boldsymbol{x}} + \overline{\boldsymbol{F}}\boldsymbol{X} + \tilde{\boldsymbol{u}}(\boldsymbol{X}), \tag{1}$$

where $\tilde{\boldsymbol{u}}$ is a *microscale displacement fluctuation field* which encodes all of the non-affine local deformations around $\overline{\boldsymbol{x}}$. In other words, $\tilde{\boldsymbol{u}}$ encodes all of the detailed, high-frequency deformations of the

microstructure geometry that are not accounted for by the large-scale deformation $\overline{F}$. The holes in a spongy material, for example, may deform more than the stiffer elastic parts; $\tilde{u}$ would define this difference in microscale deformation. See Figure 4 for an illustration.

Next, macroscale quantities are defined to be averages over their microscale counterparts [??]:

$$\overline{x} = \frac{1}{|\Omega|}\int_\Omega x(X)\, d\Omega, \tag{2}$$

$$\overline{F} = \frac{1}{|\Omega|}\int_\Omega F(X)\, d\Omega, \tag{3}$$

where Ω is the microscale reference domain with volume $|\Omega|$, and $d\Omega$ denotes integration over Ω. The theory assumes without loss of generality that $\int_\Omega X\, d\Omega = 0$, and that the macroscale quantities vary so slowly over the RVE that they are essentially constant at the microscale [?], i.e. $\overline{x}$ and $\overline{F}$ do not depend on X. Plugging (1) into (2) and (3) and applying these assumptions gives us

$$\int_\Omega \tilde{u}(X)\, d\Omega = 0, \tag{4}$$

$$\int_\Omega \nabla\tilde{u}(X)\, d\Omega = 0. \tag{5}$$

In other words, the small-scale fluctuations in translation $\tilde{u}$ and deformation $\nabla\tilde{u}$ must average out over the RVE. In computer simulations with periodic micro-structures, (4) is satisfied by fixing the barycenter of $\tilde{u}$, and (5) is commonly satisfied by requiring $\tilde{u}$ to be periodic on the boundaries [??]:

$$\tilde{u}^+ = \tilde{u}^-, \tag{6}$$

where $\tilde{u}^+$ is the value of the fluctuation field on one side of the domain, and $\tilde{u}^-$ is its value on the corresponding opposite side. Finally, we compute the homogenized energy density as the averaged total energy in the RVE

$$\overline{\Psi} = \frac{1}{|\Omega|}\int_\Omega \Psi(X)\, d\Omega, \tag{7}$$

where Ψ and $\overline{\Psi}$ are the microscale and macroscale energy densities respectively. For the purposes of simulation, we can compute forces by taking the negative gradient of this homogenized potential energy.

To restate briefly, we expand a microscale RVE from a macroscopic deformation $\overline{F}$ and with fluctuations $\tilde{u}$ that describe local deformation. We then require that the microscale deformation *on average* equals $\overline{F}$. This imposes the constraint that $\tilde{u}$ should on average not induce any additional deformation. Finally, this can be enforced through periodicity.

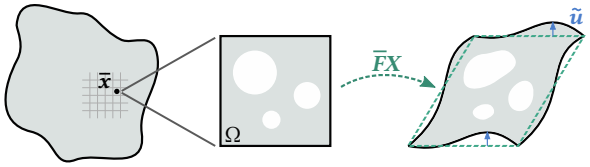


Fig. 4. At any macroscale point $\overline{x}$, we can observe a microscale RVE with reference domain Ω. The RVE is deformed through an affine transformation given by $\overline{F}$ (dashed lines) and additional periodic fluctuations $\tilde{u}$ (blue). Note that the deformation of the holes is described by a combination of $\overline{F}$ and $\tilde{u}$.

4.2 Nonlinear Homogenization of Thin Shells

Next, we apply this rationale to the problem of homogenizing a yarn-level microscale to a thin-shell macroscale. The main challenge here is to find a suitable analogy to Equation (1) that works for thin shells instead of volumes. Previous work on thin shell homogenization relies on a small curvature assumption and uses first or second order expansions for the RVE (e.g. [?]). This effectively replaces bending modes with shearing or stretching of the material, as illustrated in Figure 5. For microscale materials that resist stretching far more than bending, the erroneous stretching can introduce artificial stiffness in the homogenized response for macroscale bending. To support our goal of homogenizing highly flexible materials, this section proposes a novel non-linear thin shell expansion based on metrics from differential geometry.

On the macroscale, we have a thin shell $\overline{x}$ that is defined through its midsurface $\overline{\varphi}$, which is extruded along the normal $\overline{n}$:

$$\overline{x}(\overline{\xi}_1, \overline{\xi}_2, h) = \overline{\varphi}(\overline{\xi}_1, \overline{\xi}_2) + h\,\overline{n}(\overline{\xi}_1, \overline{\xi}_2), \tag{8}$$

where $\overline{\xi}_\alpha$ are the flat reference coordinates of the midsurface, and h is the thickness coordinate. The left side of Figure 6 illustrates this parametrization.

We locally define deformations with the first fundamental form $\overline{\mathrm{I}}$ for in-plane deformation and the second fundamental form $\overline{\mathrm{II}}$ for bending modes. With surface tangents $\overline{a}_\alpha = \overline{\varphi}_{,\alpha}$ we have

$$\overline{n} = \frac{\overline{a}_1 \times \overline{a}_2}{|\overline{a}_1 \times \overline{a}_2|}, \tag{9}$$

and we compute the components of the fundamental forms as

$$\overline{\mathrm{I}}_{\alpha\beta} = \quad \overline{a}_\alpha \cdot \overline{a}_\beta, \tag{10}$$

$$\overline{\mathrm{II}}_{\alpha\beta} = -\overline{n}_{,\alpha} \cdot \overline{a}_\beta. \tag{11}$$

We construct the RVE expansion similar to (1):

$$x(\xi_1, \xi_2, h) = \varphi(\xi_1, \xi_2) + h\, n(\xi_1, \xi_2) + \tilde{u}(\xi_1, \xi_2, h) \tag{12}$$

with microscale midsurface φ, its normal n, and fluctuation field $\tilde{u}$. In an analogy to (1), which deforms the volumetric RVE based on the macroscale quantity $\overline{F}$, we deform the thin shell RVE with a midsurface φ *derived from the macroscale fundamental forms* $\overline{\mathrm{I}}$ and $\overline{\mathrm{II}}$. The function $\varphi(\xi_1, \xi_2) + h\, n(\xi_1, \xi_2)$ applies a low-resolution spatial deformation across the entire microstructure (illustrated by the dashed line in Figure 6), while $\tilde{u}$ encodes the remaining high-frequency

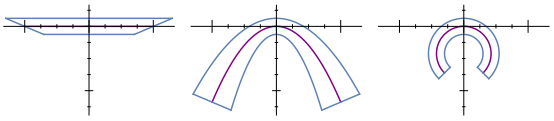


Fig. 5. Comparison of a first order (left), second order (middle) and our non-linear expansion (right) of thin-shell RVEs in a curved configuration. The lower order expansions show strong artifacts as bending modes are approximated through shearing (left) or stretching (middle).

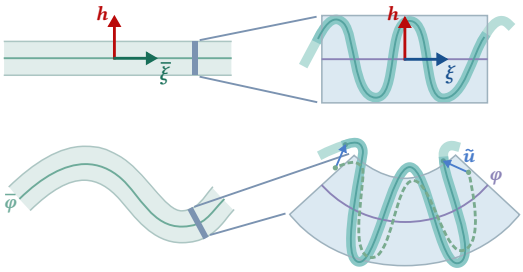


Fig. 6. A macroscopic line-segment (left) is expanded in-plane into a curved microscale volume (right). Top and bottom show reference and deformed configurations respectively. This example uses a squiggly yarn for the microstructure, and we indicate the fluctuations $\tilde{\boldsymbol{u}}$ as offsets to the yarn deformed purely from its embedding (dashed).

details of the thin-shell microgeometry. In a knitted microstructure, for example, $\tilde{\boldsymbol{u}}$ prescribes how the individual threads stretch, slide, twist, and bend relative to each other. Figure 6 illustrates a 2D schematic, and Figure 7 shows a 3D rendering of this expansion. Notice that the thickness coordinate h is shared between both micro- and macroscale since our thin-shell homogenization averages only the in-plane coordinate.

Defining the Midsurface. Our goal here is to create a midsurface $\boldsymbol{\varphi}$ in (12) with constant fundamental forms $\mathbf{I}, \mathbf{II}$ matching those of the macroscale. Although it is possible to derive such constant-fundamental-form surfaces analytically, the exact solutions are only compatible with a limited set of boundary conditions. Here, we present a more general least-squares solution to this surface-reconstruction problem.

Inspired by the rotation-strain decomposition for deformation extrapolation [?], we begin with the polar decomposition of the midsurface gradient

$$\nabla\boldsymbol{\varphi} = \begin{pmatrix}\boldsymbol{a}_1 & \boldsymbol{a}_2\end{pmatrix} = \boldsymbol{R}\bar{\boldsymbol{S}}. \tag{13}$$

Here, the 3×2 matrix $\bar{\boldsymbol{S}}$ represents the constant in-plane deformation and $\boldsymbol{R}$ is a 3×3 rotation matrix that aligns $\bar{\boldsymbol{S}}$ with the tangent plane of the curved surface. Without loss of generality, we choose the macroscale frame of reference such that $\begin{pmatrix}\bar{\boldsymbol{a}}_1 & \bar{\boldsymbol{a}}_2\end{pmatrix} = \bar{\boldsymbol{S}}$ and $\bar{\boldsymbol{n}} = \begin{pmatrix}0 & 0 & 1\end{pmatrix}^\top$. Note that $\nabla\boldsymbol{\varphi}$, $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, and $\boldsymbol{R}$ vary along the midsurface; we omit the (ξ_1, ξ_2) function notation when convenient for readability.

We want to match $\mathbf{I} = \bar{\mathbf{I}}$. With $\mathbf{I} = \nabla\boldsymbol{\varphi}^\top\nabla\boldsymbol{\varphi}$ and (13) we get

$$\bar{\boldsymbol{S}}^\top\bar{\boldsymbol{S}} = \bar{\mathbf{I}}, \tag{14}$$

allowing us to compute $\bar{\boldsymbol{S}}$ in (13) from the principal square root of the first fundamental form $\bar{\mathbf{I}}$:

$$\bar{\boldsymbol{S}} = \begin{pmatrix}\sqrt{\bar{\mathbf{I}}} \\ 0 \quad 0\end{pmatrix}. \tag{15}$$

To match $\mathbf{II} = \bar{\mathbf{II}}$, we compute $\boldsymbol{R}(\xi_1, \xi_2)$ in (13) by integrating the normal curvatures $\bar{\boldsymbol{n}}_{,\alpha}$ outward from the RVE center $\xi_1 = \xi_2 = 0$. We perform this integration with an analytic expression for the exponential map, which we explain in detail in the supplementary material (Section ??).

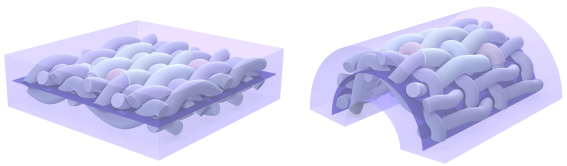


Fig. 7. A periodic yarn pattern microstructure is shown with its associated midsurface in an undeformed (left) and deformed state (right).

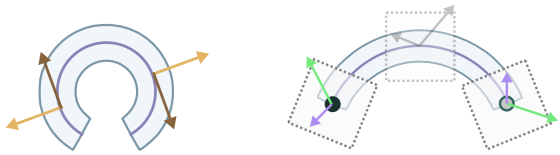


Fig. 8. (a) Naive averaging creates a null-space of growing/shrinking cylinder radii as well as sliding along the surface. As an example, the displacements indicated as pairs of orange or brown arrows would cancel each other respectively, whereas with our co-rotated averaging they are treated as the same *rotated* displacement. (b) Our co-rotated periodicity compares fluctuations (arrows) by rotating them into a common frame (gray).

Now that we know $\boldsymbol{R}$ and $\bar{\boldsymbol{S}}$, we solve (13) for $\boldsymbol{\varphi}$ in the least squares sense, giving us a vector Poisson equation with natural boundary conditions:

$$\nabla^2\boldsymbol{\varphi} = \nabla \cdot \boldsymbol{R}\bar{\boldsymbol{S}} \text{ inside the domain,} \tag{16}$$

$$\boldsymbol{N} \cdot \nabla\boldsymbol{\varphi} = \boldsymbol{N} \cdot \boldsymbol{R}\bar{\boldsymbol{S}} \text{ on the boundary.} \tag{17}$$

This equation gives the exact solution for singly-curved surfaces and can generalize to solutions for non-constant $\mathbf{I}$ and $\mathbf{II}$. We solve the system numerically by discretizing the surface as a regular grid and using standard finite differencing. This midsurface can now be used in (12) to completely describe a highly deformed thin shell microstructure, as illustrated in Figure 7.

Co-Rotated Boundary Conditions. To complete our analogy with the homogenization strategy in Section 4.1, we must derive constraints on the fluctuation field $\tilde{\boldsymbol{u}}$ which make sense for thin shells. Unfortunately, as illustrated in Figure 8a, the simple averages proposed in (4) and (5) can lead to erroneous cancellation of fluctuations when applied to a highly deformed domain, leading to undesired nullspaces in the RVE.

To address this problem, we propose to average quantities by parallel transporting them to a common frame. The rotation $\boldsymbol{R}$ from earlier rotates $\bar{\boldsymbol{n}} = \boldsymbol{n}(0, 0)$ to $\boldsymbol{n}(\xi_1, \xi_2)$ and thus describes orthogonal frames oriented along the midsurface normal. Therefore, we can use its transpose to align local frames for fluctuations, resulting in the modified constraint

$$\int_\Omega \boldsymbol{R}^\top\tilde{\boldsymbol{u}}\, d\Omega = \mathbf{0}. \tag{18}$$

With a bit more work (explained in supplementary material Section ??), we can also derive a co-rotated constraint on the derivative

of $\tilde{\boldsymbol{u}}$:

$$\int_\Omega \boldsymbol{R}^\top \tilde{\boldsymbol{u}}_{,\alpha}\, d\Omega = \boldsymbol{0}, \tag{19}$$

implying an analogous co-rotated version of (6):

$$(\boldsymbol{R}^\top \tilde{\boldsymbol{u}})^+ = (\boldsymbol{R}^\top \tilde{\boldsymbol{u}})^-, \tag{20}$$

which is satisfied by splitting the boundary of the midsurface domain Γ into opposing parts $\partial\Gamma^+$ and $\partial\Gamma^-$, and using this constraint as periodic boundary conditions. Figure 8b illustrates how our co-rotated periodicity aligns displacements.

Finally, for macroscale thin shell simulations, we are interested in homogenizing an elastic energy *area* density. Instead of dividing the total energy by the volume as in (7), we divide by the area of the RVE midsurface to get

$$\overline{\Psi} = \frac{1}{|\Gamma|}\int_\Omega \Psi(\xi_1, \xi_2, h)\, d\Omega, \tag{21}$$

with $|\Gamma|$ being the area of the midsurface domain.

To summarize, we are now able to take a macroscale deformation given by $\bar{\mathbf{I}}$ and $\overline{\mathbf{II}}$ and compute a midsurface $\boldsymbol{\varphi}$ from (16)–(17). This defines the fluctuation field $\tilde{\boldsymbol{u}}$ through (12), on which we can then enforce the translation and periodicity constraints (18) and (20), and compute homogenized energy area densities $\overline{\Psi}$ with (21).

5 YARN PATTERN SIMULATION

For any periodic yarn pattern, we aim to compute a mapping from deformation to homogenized energy densities. To minimize the dimensionality of the problem, we seek the energy at the elastostatic equilibrium configuration, subject to the macroscopic deformation. The elastostatic assumption is common in many applications like animation [**?**], fracture simulation [**?**], and structural optimization [**?**], because it captures the overall behavior of a material without needing to compute dynamic effects. In our case, this equilibrium state corresponds to the physical state with yarn collisions resolved and the yarns being at rest with respect to bending, twisting, and stretching.

To deform a microscale periodic yarn patch, we embed it into the RVE as shown in Figure 7. Finding the elastostatic equilibrium amounts to a constrained optimization problem of minimizing the homogenized energy with respect to the fluctuations $\tilde{\boldsymbol{u}}$ and subject to the translation and deformation constraints; i.e.,

$$\overline{\Psi} = \min_{\tilde{\boldsymbol{u}}} \frac{1}{|\Gamma|}\int_\Omega \Psi(\tilde{\boldsymbol{u}})\, d\Omega \quad \text{s.t. (18) and (20).} \tag{22}$$

Figure 9 shows a yarn pattern before and after relaxing it into its optimized state.

The number of tiles within an RVE is a choice that determines which scales of buckling are handled by homogenization, and which ones are handled by the cloth simulator. In this work, we chose to use a small RVE size for each pattern primarily based on computational cost, and have not explored larger sizes. We leave the study of RVE sizes and buckling frequencies as future work.

5.1 Yarn Model

We simulate yarns using discrete elastic rods (DER) [**??**] with the yarn-level cloth collision forces of [**?**] modified for linear spline segments. Real wool yarns consist of many threads wound together, so they may resist bending and twisting much less than stretching. To add more flexibility to our yarn simulations, we therefore add an additional parameter γ to scale bending and twisting energies in relation to stretching energy. Thus, we compute the integral in (22) as the sum of stretching E_s, bending E_b, twisting E_t, and collision energies E_c of yarns in the periodic patch:

$$\int_\Omega \Psi\, d\Omega = E_s + \gamma E_b + \gamma E_t + E_c. \tag{23}$$

For the definition of the individual energies, see [**?**] for E_c and [**?**] for the other terms. As discussed in Section 3, we omit inter-yarn friction in the micro-scale quasistatic optimization.

The elastic energy terms in this model require that we know the rest shape of each yarn. Because the act of knitting and weaving can actually change the rest shape of a yarn (as seen in Figure 10), obtaining it is a non-trivial task. In our experiments, we apply a heuristic that the rest pattern should be in equilibrium relative to the stretching energy; inspired by **?**, we apply tension by shortening the yarns' rest lengths, and then we shrink the periodic lengths of the pattern to find an energy minimum relative to stretching. We explain this initialization process in detail in the supplementary material (Section **??**).

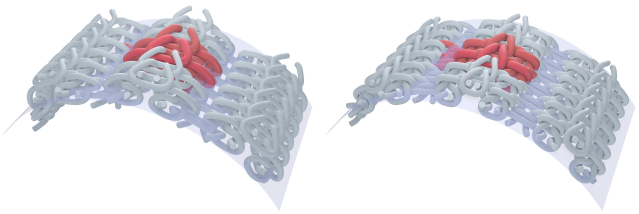


Fig. 9. We show a stretched rib pattern before (left) and after (right) optimization with the periodic tile highlighted in red, periodic ghost segments in gray, and the midsurface in translucent blue. Notice how the rib pattern's yarn loops naturally tighten under tension while maintaining the curvature of the surface.

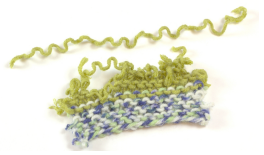


Fig. 10. In this real-world example, we extracted the top strand of wool yarn from the knit pattern below, and allowed the yarn to come to rest. The yarn clearly has a bent rest shape related to the pattern it was knitted into.

5.2 Periodicity

Yarns on one side of the patch can interact with yarns on the opposite side through periodic collisions or by being periodically connected. Therefore, we have to consider periodic discrete elastic rod and collision forces. We introduce *ghost segments* that copy and tile the yarns along the periodic field $\boldsymbol{R}^\top \tilde{\boldsymbol{u}}$ implied by the constraint (20). Ghost segments do not contribute to the energy in the (23); they

simply copy the motion of the primary yarns and act as colliders and boundaries for the yarn segments in the RVE. Figure 9 shows these ghost segments colored gray.

In addition to positional degrees of freedom, [?] incorporates material frames and edge twists. We enforce periodicity on reference frame directors $\underline{\boldsymbol{d}}_\alpha$ and twist variables θ via

$$(\boldsymbol{R}^\top \underline{\boldsymbol{d}}_\alpha)^+ = (\boldsymbol{R}^\top \underline{\boldsymbol{d}}_\alpha)^-, \tag{24}$$

$$\theta^+ = \theta^-, \tag{25}$$

where + and − denote an original and copied edge respectively.

5.3 Homogenization Constraints

For the purposes of homogenization, we have to impose the translation constraint (18), periodic vertex positions (20), and periodic edge twists (25) on the microscale. Additionally, the yarn forces are invariant to a constant twist, so we remove this nullspace by requiring the total twist per periodically connected yarn to be zero. We found that the reference frames do not drift from their constraint manifold (24) over time, so we do not actively enforce this constraint after initialization.

We enforce the periodicity constraints by eliminating the copied degrees of freedom from the linear system in the Newton step. Exploiting the fact that any periodic vertex or twist relates linearly to exactly one other vertex or twist through (20) and (25), we can define reduced degrees of freedom $\boldsymbol{y}$ through

$$\tilde{\boldsymbol{C}}\boldsymbol{y} + \tilde{\boldsymbol{d}} = \boldsymbol{q}, \tag{26}$$

where $\boldsymbol{q}$ is the vector of all vertex positions and edge twists. Notably, $\tilde{\boldsymbol{C}}$ is sparse and will preserve the sparsity of the Newton system. This elimination of variables is based on parametrizing the nullspace of all periodicity constraints. We discuss its construction in Appendix A.

On the other hand, due to its density, enforcing the translation constraint (18) by parametrizing its nullspace would result in a dense $\tilde{\boldsymbol{C}}$. Instead, we enforce this constraint with Lagrange multipliers. In addition, we also use Lagrange mutlipliers to remove the nullspace of constant twists along a yarn by requiring

$$\sum_i \theta_i = 0 \tag{27}$$

for each (periodically connected) yarn. We concatenate the translation and twist constraints to get

$$\boldsymbol{C}_L\boldsymbol{q} = \boldsymbol{d}_L. \tag{28}$$

5.4 Optimization Step

We can now solve the constrained minimization problem in (22). Using Newton iteration, each step to solve for increments $\delta\boldsymbol{y}$ and Lagrange multipliers $\boldsymbol{\lambda}$ is given by

$$\begin{pmatrix} \tilde{\boldsymbol{C}}^\top \boldsymbol{H}\tilde{\boldsymbol{C}} + \alpha\boldsymbol{I} & \tilde{\boldsymbol{C}}^\top {\boldsymbol{C}_L}^\top \\ \boldsymbol{C}_L\tilde{\boldsymbol{C}} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{y} \\ \boldsymbol{\lambda} \end{pmatrix} = - \begin{pmatrix} \tilde{\boldsymbol{C}}^\top \nabla E \\ \boldsymbol{C}_L\boldsymbol{q} - \boldsymbol{d}_L \end{pmatrix}, \tag{29}$$

where E is the total energy, $\boldsymbol{H} = \frac{\partial^2 E}{\partial \boldsymbol{q} \partial \boldsymbol{q}}$ is its Hessian, and and α is an exponentially decaying regularizer to help convergence. We also limit the maximal vertex displacement per step to a fraction of a yarn radius to avoid missing collisions between iterations, and we observed improved numerical conditioning if we rescale positional degrees of freedom relative to twists. We provide these details, as well as initialization and stopping criteria for this optimization algorithm in the supplementary material (Section ??).

6 FITTING

At this point, we are able to compute an energy density $\overline{\Psi}$ for a yarn pattern given an input deformation $\overline{\mathbf{I}}, \overline{\mathbf{II}}$. Our next step is to build a database of entries sampling this $\overline{\Psi}(\overline{\mathbf{I}}, \overline{\mathbf{II}})$ function, and then approximate the data by fitting a model to it. However, the energy landscape can be noisy due to multiple microscale equilibria — the yarn pattern can buckle, interacting yarns are generally multistable and slide over each other. Especially in compressive regimes, the pattern can buckle differently for similar strains, leading to noise in the energies. Local minima in the fit then introduce noisy restshapes and popping in the final macroscale simulation (see Figure 11). Additionally, our data is neither convex nor is it well-fit by polynomials. After experimenting with several fitting schemes, we settled on the strategy of first regularizing the input data, and then fitting a model as a sum of regularized splines while enforcing quasiconvexity and piecewise monotone interpolation. We will discuss the main ideas of the fitting procedure in this section, and we provide further details in our supplementary material (Section ??).

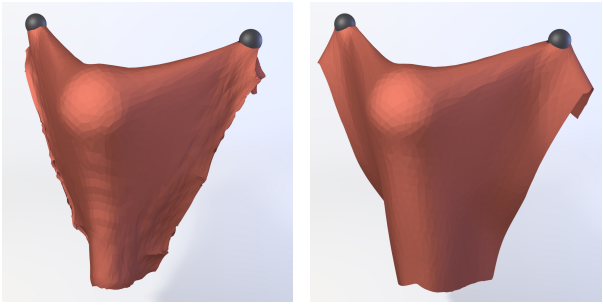


Fig. 11. Insufficient regularization can negatively affect simulated rest shapes. Here, a draped rib knit shows noisy boundaries (left) compared to a fit with better regularization (right).

6.1 Parametrization and Sampling

We begin by choosing a reparametrization of the input strains $\overline{\mathbf{I}}$ and $\overline{\mathbf{II}}$ that is better suited to sampling and interpolation. We desire each input parameter to be valid over a fixed interval independent of other parameter values, so that we can use standard interpolation schemes over rectilinear grids. Furthermore, we wish to avoid sampling over the full n-dimensional space of possible strains, but still capture pairwise interactions such as the Poisson's ratio, influence of stretching on bending, and so on.

To start, we reparametrize the in-plane strains. Using the entries of $\overline{\mathbf{I}}$ is problematic as its off-diagonal entry $\boldsymbol{a}_1 \cdot \boldsymbol{a}_2$ not only encodes the shearing angle but is also influenced by the lengths of the $\boldsymbol{a}_\alpha$. Instead, we define weft-stretching s_x, shearing s_a, and warp-stretching

s_y strains as

$$s_x = \sqrt{\bar{\mathrm{I}}_{11}} - 1, \qquad s_a = \frac{\bar{\mathrm{I}}_{12}}{\sqrt{\bar{\mathrm{I}}_{11}\bar{\mathrm{I}}_{22}}}, \qquad s_y = \sqrt{\bar{\mathrm{I}}_{22}} - 1, \tag{30}$$

and the combined in-plane strain $\boldsymbol{s} = \begin{pmatrix} s_x & s_a & s_y \end{pmatrix}^\top$. Here, we use the terms "weft" and "warp" to refer to the directions ξ_1 and ξ_2 respectively.

The difficulty with the bending strain $\overline{\mathbf{II}}$ is that it is not possible to construct a microscale patch with constant strain unless it is singly curved, i.e. the rank of $\overline{\mathbf{II}}$ is ≤ 1. We were further unable to find a satisfactory parametrization for the space of all singly-curved bending strains. Instead, we choose to only sample the response to bending along two orthogonal directions. That is, we collect one set of data with $\overline{\mathbf{II}}$ of the form $\mathrm{diag}(\lambda_x, 0)$, and another set with $\overline{\mathbf{II}} = \mathrm{diag}(0, \lambda_y)$.

The data then represent samples of the function along two subspaces: one with arbitrary $\boldsymbol{s}$ and bending only in x, and one with arbitrary $\boldsymbol{s}$ and bending only in y. As described the next section, we interpolate the data in each subspace to obtain fits $\overline{\Psi}_x(\boldsymbol{s}, \lambda_x)$ and $\overline{\Psi}_y(\boldsymbol{s}, \lambda_y)$. Finally, we describe how to interpolate between them to define the fitted energy density for arbitrary bending strain $\overline{\mathbf{II}}$.

Note that our choice of axis-aligned bending and stretching corresponds to the weft and warp directions that are dominant in the patterns we investigate, but in general the orientation of the bases is arbitrary.

Prior to fitting, we normalize all strains $(s_x, s_a, s_y, \lambda_x, \lambda_y)$ by their maximum absolute values in the data, which ensures that stretching and bending strains are treated as equally important. We have tried various strategies to mitigate the noise in the data induced by buckling, including prohibiting specific buckling modes through constraints and even penalizing yarn motion normal to the midsurface. However, we were unable to eliminate noise without affecting the overall elastic response and concluded that homogenization of micro-scale buckling is a difficult problem. As a first step, we settled on regularizing the data by re-sampling it using moving least squares interpolation.

6.2 Fitting and Interpolation

We define a fitting procedure for multidimensional data which captures pairwise interactions between parameters without requiring high-dimensional sampling. Consider a function f depending on many parameters $\theta_1, \theta_2, \ldots$. Inspired by **?**, we additively split it into the form

$$f(\theta_1, \ldots, \theta_n) = f_0 + \sum_i f_i(\theta_i) + \sum_{i<j} f_{ij}(\theta_i, \theta_j). \tag{31}$$

Without loss of generality, we may fix $f_i(0) = 0$ and $f_{ij}(0, \theta_j) = f_{ij}(\theta_i, 0) = 0$. Thus the one-dimensional term f_i encodes the response to θ_i holding other parameters at zero, and the two-dimensional term f_{ij} encodes the *residual* response to both θ_i and θ_j, i.e. the component of $f(\ldots, \theta_i, \ldots, \theta_j, \ldots)$ not explained by $f_0 + f_i(\theta_i) + f_j(\theta_j)$.

Therefore, the f_{ij} terms describe cross-modal material responses, including stretching in two directions or simultaneous stretching and bending. Notably, our homogenization method is capable of sampling these cross-modal deformations.

To fit the components of (31), we measure $f_0 = f(0, 0, 0, \ldots)$, we fit the one-dimensional f_i terms using piecewise monotone cubic splines [**?**], and we fit the two-dimensional residual f_{ij} terms using our novel extension of **?** to spline patches. We also apply a heuristic outward marching algorithm to ensure quasiconvexity. **?** enforce convexity in their fits. However, we found that this would not describe our data well, and we opted for quasi-convexity as the closest choice. This strategy makes our results stable (see Figure 11) with the downside of not perfectly fitting the data. Outside of the sampled range, we linearly extrapolate the fitted splines. We provide details for each of these steps in our supplementary material (Section **??**). Figure 12 shows data and fit for the 1D splines. Figure 13 compares data, 1D fits, 2D residuals, and the cumulative fit.

Our method makes the simplifying assumption that there are only pairwise interactions between parameters. What this assumption buys us is a dramatic economy of sampling: even for arbitrarily high-dimensional parameter spaces, our procedure only needs samples along coordinate axes and 2D coordinate planes. When the assumption is violated, however, our approach may not preserve convexity. For example, $f(x, y, z) = \max(x^2, y^2, z^2)$ is a convex function for which our fit is nonconvex.

The above procedure is applied to the singly-curved data $\overline{\Psi}_x$ and $\overline{\Psi}_y$ defined previously. Of course, the zero-curvature data points and the 1D and 2D fitting terms not involving curvature will be shared between both. Finally, to define our fitted energy density for an arbitrary curvature $\overline{\mathbf{II}}$, we look at the eigenvalues of $\overline{\mathbf{II}}$, λ_1 and λ_2, and the squared cosine c^2 of the angle between the eigenvector corresponding to λ_1 and the x-axis. In our supplementary document (Section **??**), we show how to robustly compute these values. Now we define $\overline{\Psi}(\boldsymbol{s}, \overline{\mathbf{II}})$ as

$$\begin{aligned} \overline{\Psi}(\boldsymbol{s}, \overline{\mathbf{II}}) = c^2 &\left(\overline{\Psi}_x(\boldsymbol{s}, \lambda_1) + \overline{\Psi}_y(\boldsymbol{s}, \lambda_2) \right) \\ &+ (1 - c^2) \left(\overline{\Psi}_x(\boldsymbol{s}, \lambda_2) + \overline{\Psi}_y(\boldsymbol{s}, \lambda_1) \right). \end{aligned} \tag{32}$$

Limitations. We found the fitting problem particularly challenging due to the complex interactions between deformation modes, the numerical noise in the data, and especially the sensitivity of macroscale simulations to local minima in the energy density (Figure 11). We invested a great deal of effort to design a fitting scheme that works well for all the yarn patterns we tested, but we found a few cases unavoidable, which we summarize below. Firstly, to ensure a decent fit for the "stockinette" pattern, which features a strong tendency to curl, we found it necessary to concentrate spline control points for 2D residual terms involving bending strains more closely around the origin (Figure 14), and to apply a higher quasiconvexity parameter in the marching step. We believe that this may be caused by the far-off-center bending minimum and the additively split model thus creating local minima. Secondly, we observed yarn-level reference simulations to exhibit symmetric rest shapes with zero shear; to ensure that this behavior is preserved in our macroscale simulations, we symmetrized our data with respect to s_a. Finally, we disabled our heuristic quasiconvexity marching for the two-dimensional $f_{13}(s_x, s_y)$ term, which would otherwise prevent us from modeling Poisson's ratio. We refer the reader to the supplementary material for the full details of the fitting algorithm,

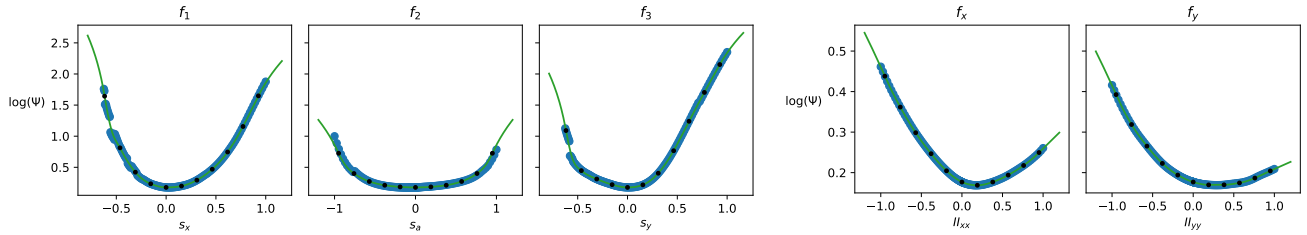


Fig. 12. One-dimensional in-plane (left) and bending (right) terms for the honeycomb pattern. We show data in blue, the fit as a green line, and spline control points as black dots. Notice the off-center minimum for the bending terms, which corresponds to the pattern's curved rest shape.

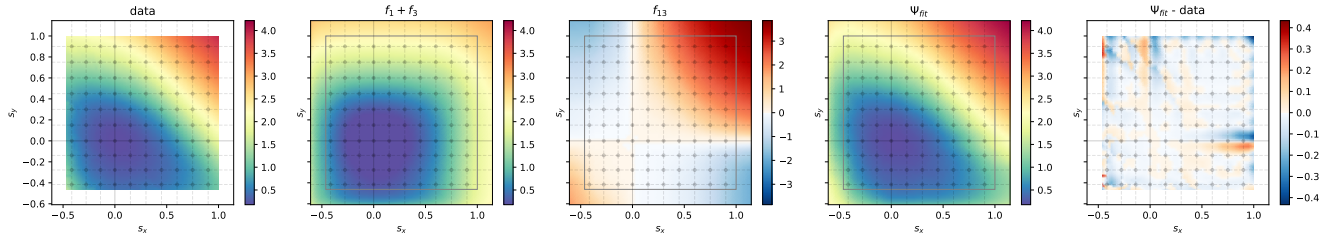


Fig. 13. Fit of the 2D term $f_{13}(s_x, s_y)$ for the honeycomb pattern. From left to right: data, sum of 1D fits, 2D residual, cumulative fit, and fitting error. The colors show the magnitude on a symmetric-log scale, and we show extrapolated values outside of the data range (indicated as a rectangle). Note that this term shows the area preservation of the material; while increasing tension along one axis, the minimum along the other moves towards a compressed state. Crucially, this behavior is missing from just the sum of 1D terms.

including the above modifications, as well as the raw strain-energy data and our fitting code.

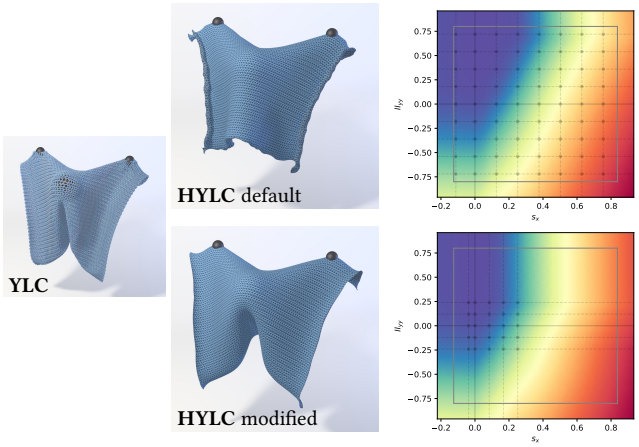


Fig. 14. We show a simulated restshape and the plot for a representative bending residual for the homogenized stockinette with default and modified spline control points, and a yarn-level reference (YLC). Even though the fit is smooth, default control point locations create artifacts at the fabric boundary.

7 CLOTH SIMULATION

We now want to drive a thin-shell cloth simulator using the continuum models fit in the previous section. The cloth is discretized as a triangle mesh, which represents the macroscale thin shell midsurface $\overline{\boldsymbol{\varphi}}$. Similarly, we need to discretize $\overline{\mathbf{I}}$ and $\overline{\mathbf{II}}$ to compute in-plane and bending strains, (30) and $\lambda_1, \lambda_2, c^2$, on the triangle mesh. For robust simulation, we use implicit integration, which requires computing the Hessian of the energy. To improve stability, we enforce positive definiteness in the Hessian. Dynamic yarn friction is partially modeled via Rayleigh damping in the continuum simulations, but we leave the inclusion of friction into the homogenization procedure as future work.

For each triangle, we first compute its deformation gradient

$$\boldsymbol{F}_\triangle = \left(\overline{\boldsymbol{\varphi}}_1 - \overline{\boldsymbol{\varphi}}_0, \overline{\boldsymbol{\varphi}}_2 - \overline{\boldsymbol{\varphi}}_0\right)\left(\overline{\boldsymbol{\xi}}_1 - \overline{\boldsymbol{\xi}}_0, \overline{\boldsymbol{\xi}}_2 - \overline{\boldsymbol{\xi}}_0\right)^{-1}, \tag{33}$$

where $\overline{\boldsymbol{\varphi}}_j$ and $\overline{\boldsymbol{\xi}}_j$ are the world-space and material-space coordinates of vertex j, and the *triangle-averaged shape operator* [?]

$$\boldsymbol{\Lambda} = \sum_i \frac{\theta_i}{2Al_i} \boldsymbol{t}_i \otimes \boldsymbol{t}_i, \tag{34}$$

where θ_i is the signed angle between this and the i-th neighboring triangle's normals, A is the triangle area, l_i are edge lengths, and $\boldsymbol{t}_i$ are vectors of length l_i perpendicular to each edge and the inner triangle normal. All quantities in (34) are computed in *world-space*. With this, we compute the discrete fundamental forms as

$$\overline{\mathbf{I}}_\triangle = \boldsymbol{F}_\triangle{}^\top \boldsymbol{F}_\triangle, \tag{35}$$

$$\overline{\mathbf{II}}_\triangle = \boldsymbol{F}_\triangle{}^\top \boldsymbol{\Lambda}\, \boldsymbol{F}_\triangle. \tag{36}$$

Because of (34), the degrees of freedom involved in a triangle's strain also include the triangle vertices of up to three neighboring triangles. Denoting the combined degrees of freedom as $\boldsymbol{q}_\triangle$ and

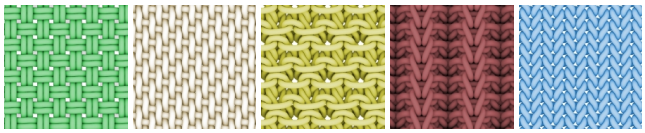


Fig. 15. The patterns used in our results with abbreviated names.

the collected strains $\boldsymbol{z} = (s_x, s_a, s_y, \lambda_1, \lambda_2, c^2)$, the total energy of a triangle is given by

$$E_\triangle = A\,\overline{\Psi}(\boldsymbol{z}(\boldsymbol{q}_\triangle)). \tag{37}$$

Since our energies are nonconvex, their Hessians are not guaranteed to be positive definite, which negatively affects stability. Inspired by **?**, we enforce positive definiteness by clamping the eigenvalues of per-triangle sub-Hessians $\frac{\partial^2 E_\triangle}{\partial \boldsymbol{q}_\triangle \partial \boldsymbol{q}_\triangle}$ to be non-negative using an eigensolver for self-adjoint matrices in the library Eigen [**?**]. The global system in the implicit timestep will then be positive definite as a sum of the positive semi-definite sub-Hessians and the positive definite global mass matrix.

8 RESULTS

To summarize, our pipeline first takes in a periodic yarn pattern and elastic rod material properties, simulates the pattern subject to various deformed boundary conditions, and records the resulting potential energy density. We then create a data-driven strain-parameterized material model for each yarn pattern and simulate the material in an existing thin shell finite element solver (ArcSim [**??**]). We include pseudo-code for each step in our pipeline in a supplementary document.

In our experiments, we wanted to model a variety of yarn patterns with notably different topologies and macroscale material effects. We drew several patterns from the yarn pattern database of **?** (`basket2_2`, `satin2_3`, `slip_stitch_honeycomb`, and `cartridge_belt_rib`), and implemented a custom stockinette knit pattern of our own. Figure 15 shows the five patterns. The knitted patterns are topologically quite different from each other and from the woven patterns, leading to significant variance in macroscopic effects like area preservation, resistance to stretching, and out-of-plane curling. We rescale the patterns to have a yarn radius of 1 mm and smaller variants of the satin and stockinette patterns to 0.1 mm. Table 1 lists the yarn-level parameters for each pattern; we choose parameters to achieve realistic-looking yarn-level simulations.

We render cloth simulated with our models using ambient occlusion and normal map textures, which we create by projecting the periodic yarn patterns. Thus, our results cannot easily generate visible gaps between yarns as seen in Figure 14, regardless of the quality of the homogenization. It is possible to drive the deformation of detailed yarn-level geometry using the coarser, simulated mesh, although such a strategy may be computationally expensive for large garments or small knits (e.g. Figure 18). We attribute the differences between yarn-level and homogenized results in Figure 14 to both texture mapping as well as an imperfectly homogenized model.

Table 1. Yarn-level parameters per pattern including Young's modulus E, the shear modulus G, the bending and twisting stiffness multiplier γ, the collision stiffness k_{contact}, and the density ρ.

Pattern	E (Pa)	G (Pa)	γ	k_{contact} $\left(\frac{\text{kg}}{\text{s}^2}\right)$	ρ $\left(\frac{\text{kg}}{\text{m}^3}\right)$
basket	1e5	4e4	0.1	1.2e1	1.2e2
honey	5e5	2e5	0.1	6e1	1.2e2
rib	5e5	2e5	0.001	6e1	6e1
satin	1e5	4e4	0.1	1.2e1	1.2e2
stock.	5e5	2e5	0.001	6e1	1.2e2
satin small	1e6	4e5	1	1e2	1.2e3
stock. small	1e6	4e5	1	1e2	1.2e3

8.1 Validation

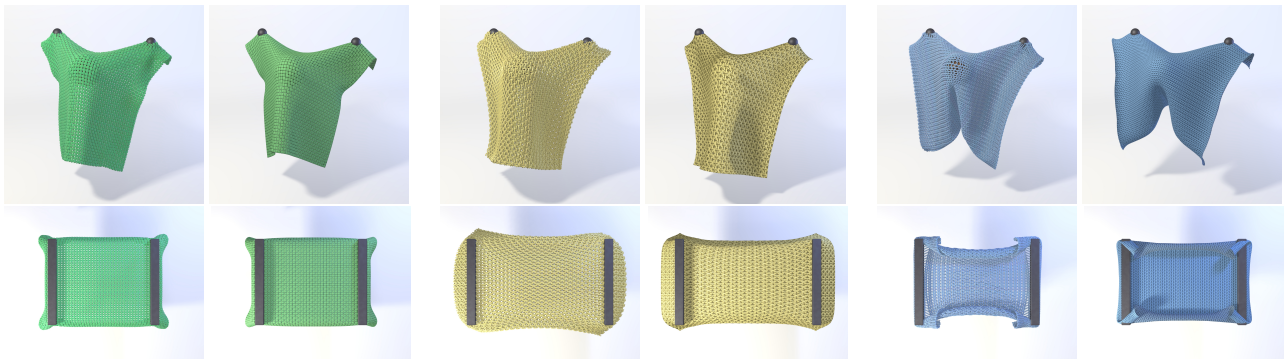
To validate our homogenized macro-material models, we run side-by-side comparison simulations between our macro-material cloth simulator and a brute-force yarn-level cloth simulator. We compare the behavior of a 30 cm × 30 cm square patch of material when stretched in different directions and draped over a spherical obstacle. Some of these comparisons are displayed in Figure 16 and Figure 1, and all of them are included in our supplementary data.

Our homogenized yarn-level cloth models generally agree well with the yarn-level cloth simulations, even though the various yarn patterns behave very differently from each other: the woven materials tend to be stiffer and exhibit no tendency to preserve area when stretched; the rib knit exhibits fairly extreme anisotropy when stretching; the stockinette stitch curls up on the boundaries when stretched or left to hang freely.

For the yarn-level simulations in this comparison, we used the non-rigid motion damping of **?**. Because our material models are based on elastic properties of the cloth, we did not yet attempt to learn damping properties. Instead, we used the continuum Rayleigh damping model implemented in ArcSim, which we tuned to empirically match the yarn-level damping model.

Note that our material models are extracted from periodic yarn patterns, so they should be able to adequately reproduce the behavior of a yarn-level simulation near the interior of the cloth. However, knitted garments generally have different stitches or fasteners near boundaries, which disrupts this periodic structure; indeed, to model boundary effects in our yarn-level simulator, we effectively "glue" the yarns together with springs that are pre-stretched in the thickness direction. These boundary effects were not included in our periodic homogenization, so we do not expect our material to behave perfectly near boundaries. Nevertheless, our results do show relatively similar boundary behaviors to the yarn-level examples.

To illustrate the merits of our multi-dimensional fitting procedure described in Section 6, we also compared our method's behavior with and without two-dimensional energy terms. As seen in Figure 17 and our supplementary data, the materials with only one-dimensional stress response do a reasonable job of approximating the overall stretching and bending resistance, but they fail to capture more complex two-dimensional compensations. Notably, the 1D models cannot capture Poisson-like behaviors, where stretching in one direction causes the material to compress in the other.



(a) basket weave (b) honeycomb (c) stockinette

Fig. 16. Comparison of direct yarn-level simulation (YLC) to simulation with our homogenized continuum models (HYLC) for drapes and stretching tests of three patterns. Our method is able to capture a wide array of phenomena such as the Poisson's ratio of the honeycomb pattern, or the more exotic restshape and curling under tension of the stockinette pattern at a fraction of the cost.

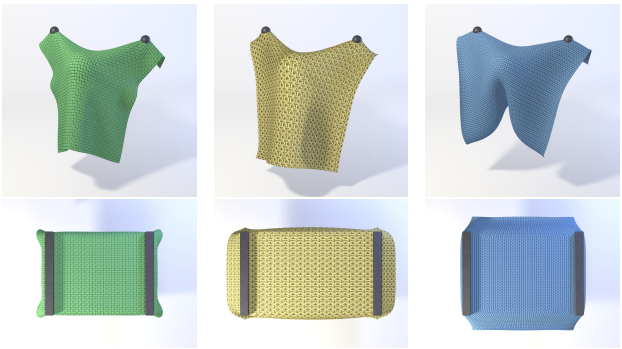


Fig. 17. Only fitting the one-dimensional energy terms for the same models as in Figure 16 shows that overall draped shapes are still captured nicely and are arguably faster and easier to fit. However, area-preservation effects and curling under tension are modeled by two-dimensional terms and as a result are lost in the simpler model.

Performance. The computational complexity of a yarn-level cloth simulator scales with the number of yarn segments. In contrast, the performance of our macroscale material scales with the number of elements in a cloth simulator, multiplied by the cost of evaluating our potential energy function (or its gradient). Yarn-level simulations also invest computational resources into carefully handling persistent inter-yarn collisions, either through small time steps or more clever collision handling. Our method deals with those persistent contacts in its preprocessing phase, and only deals with large-scale self-collisions within the cloth solver. Because our method sidesteps most of the performance bottlenecks in a yarn-level cloth simulator, we expect our method to achieve a large speedup over a yarn-level cloth, especially when the yarn density is high. Additionally, using an implicit cloth solver allows us to take larger timesteps compared to the explicit yarn-level solver, where computing Hessians becomes infeasible. Although these side-by-side examples use a modest number of yarn segments, our simulator shows significant speedups from ×3.3 to up to ×46, as seen in Table 2. Across the patterns, sampling the data for fitting takes from 15min to 76min, and the fitting itself takes less than a minute, further highlighting the cost benefit of precomputing inexpensive simulations. The stockinette examples in Table 2 have a higher "sec/frame" and number of vertices due to finer adaptive remeshing needed for resolving tight curls.

Our proposed constitutive model depends on the second fundamental form and thus requires more computation compared to standard bending models based on dihedral angles such as [**?**].

8.2 Large-scale Simulations

Because our homogenized material's computational complexity is now independent of the number of yarns, we are able to approximate the behavior of large garments with a high density of yarns. Figure 19 and our supplementary data show draped cloth simulated with models of stockinette and satin patterns rescaled to 10% of their original size. The stitch density of these materials is one hundred times higher than those we were able to feasibly simulate with a yarn-level simulator, so we do not have any direct performance or behavioral comparisons to report here.

Similarly, we are able to simulate large garments such as sweaters and shirts (Figure 1, Figure 18). We note that these homogenized knitted materials retain their unique material properties, like stretchiness (honey), anisotropic effects (rib), or curling at the boundaries (stockinette), despite the fact that they were simulated with a continuum-mechanics based cloth solver. For comparison, a direct yarn-level simulation of a stockinette sweater would require over 1.7 million vertices, compared to the 76 thousand vertices in our yarn-level validation tests. The small-stockinette shirt would require 36 million vertices.

Because our homogenized materials rely on triangle meshes instead of knitted patterns to determine their geometry, it is straightforward to simulate garments with more exotic shapes using our method (Figure 1, Figure 20). We report the simulation timings for each of these results in Table 3.

Table 2. Simulation timings for the comparisons of direct yarn-level simulation (YLC) and with our method (HYLC). Pattern names are abbreviated. All tests were performed for two orientations of the cloth (original and 90° rotation), and their videos can be found in the supplementary data. Δt denotes the timestep in seconds. sec/step denotes the average seconds per timestep. sec/frame denotes the average seconds per frame for a reference framerate of 30fps. # Vertices denotes the number of vertices in yarn-level simulations, and the average number of vertices for thin-shell simulations, which are subject to remeshing.

Simulation		HYLC				YLC			
		Δt	sec/step	sec/frame	# Vertices*	Δt	sec/step	sec/frame	# Vertices
basket drape	Fig. 16	2.09e-04	0.46	73.46	2276	1e-05	0.13	(×5.9) 430.43	65188
basket drape 90°		2.09e-04	0.46	72.74	2229	2e-05	0.23	(×5.2) 381.50	65188
basket stretch	Fig. 16	2.09e-04	0.50	80.58	2668	1e-05	0.17	(×7.0) 560.43	65188
basket stretch 90°		2.09e-04	0.52	82.96	2657	1e-05	0.24	(×9.6) 797.10	65188
honey drape	Fig. 16	1.67e-04	0.41	81.91	2091	1e-05	0.36	(×14.7) 1206.40	118140
honey drape 90°		1.28e-04	0.42	109.65	2127	1e-05	0.39	(×12.0) 1314.72	118140
honey stretch	Fig. 16	1.67e-04	0.49	98.67	2370	1e-05	0.31	(×10.3) 1017.60	118140
honey stretch 90°	Fig. 1	1.67e-04	0.44	87.50	2376	1e-05	0.29	(×10.9) 954.20	118140
rib drape		2.09e-04	0.48	76.86	2337	5e-06	0.39	(×34.2) 2625.12	157592
rib drape 90°		2.09e-04	0.48	77.32	2374	5e-06	0.53	(×46.0) 3559.10	157592
rib stretch		2.09e-04	0.48	76.35	2577	5e-06	0.38	(×33.3) 2542.47	157592
rib stretch 90°		2.09e-04	0.47	75.07	2541	5e-06	0.45	(×39.6) 2971.27	157592
satin drape	Fig. 1	2.09e-04	0.48	77.21	2297	1e-05	0.56	(×24.0) 1855.08	95040
satin drape 90°		2.09e-04	0.47	74.74	2246	1e-05	0.56	(×24.9) 1861.55	95040
satin stretch		2.09e-04	0.44	70.82	2500	1e-05	0.35	(×16.6) 1176.50	95040
satin stretch 90°		2.09e-04	0.50	79.66	2684	1e-05	0.30	(×12.4) 985.17	95040
stock. drape	Fig. 16	2.09e-04	0.96	152.96	3390	1e-05	0.19	(×4.2) 643.08	76156
stock. drape 90°		2.09e-04	1.03	165.04	3383	4e-06	0.08	(×4.0) 652.35	76156
stock. stretch	Fig. 16	2.09e-04	1.15	184.17	4415	1e-05	0.18	(×3.3) 615.83	76156
stock. stretch 90°		2.09e-04	0.79	126.91	3869	4e-06	0.08	(×5.4) 684.30	76156

basket satin honeycomb rib stockinette small satin small stock.

Fig. 18. We demonstrate the effectiveness and the rich behavior of our homogenized models for all of our patterns on simulations of sweaters and t-shirts. This freeze frame highlights: stronger stretching resistance of woven fabric (basket and satin), the anisotropy of the rib, curling of the stockinette, and the folds of the small-scale patterns.

9 CONCLUSION

This paper proposes a method for computing homogenized models capable of simulating yarn-level effects in a thin-shell cloth solver. Through homogenization of a non-linear shell, we are able to compute homogenized responses of periodic yarn patterns to macroscale deformations. We can then fit a regularized continuum model without the need for expensive measurement equipment. We compare our results with brute force simulations for multiple patterns on a series of stretching and draping tests. Our method is able to capture the rich properties of knitted fabric such as general stretching and bending anisotropy, including Poisson's ratio, while being an order of magnitude faster even on moderate scales.

Limitations and Future Work. Our model is able to abstract the yarn-level interactions into an elastic continuum; however, this

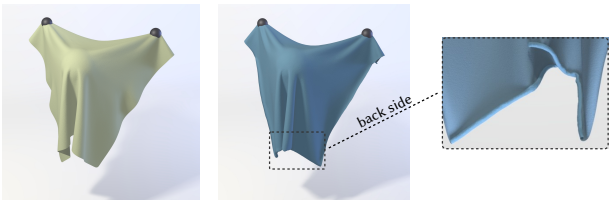


Fig. 19. Models homogenized from higher density variants of the satin (left) and stockinette (right) patterns at a 10% scale naturally produce more folds when draped. Notably, the small stockinette shows small curls on the inside, similar to cut t-shirts. Besides the scale of the folds, the larger stitch density does not affect the performance of our method.

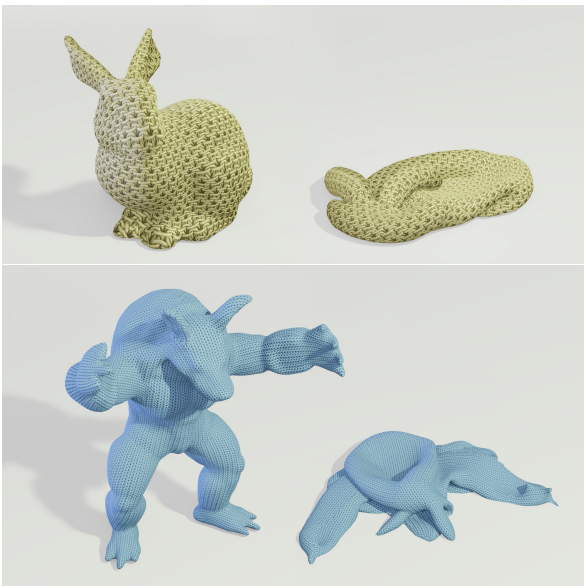


Fig. 20. Before and after of a bunny and a *yarnmadillo* simulated with our models.

implies that we do not model localized effects such as tearing or pulling on single yarns. To this end, combining our continuum model with localized yarn simulation is worth investigating.

While our model captures elastic rest shapes well, we ignore yarn-level friction and hysteresis in our homogenization procedure. Although our method can be combined with other macroscale damping and plasticity models, we would like to explore homogenizing viscous and plastic effects from yarn-level simulations as well. We have left cloth-cloth and cloth-obstacle frictional contact entirely to the continuum solver; the more recent Argus simulator [?] could be used in place of ArcSim for improved accuracy there.

Homogenization theory assumes a small RVE compared to the macroscale deformation. Although our co-rotated boundary conditions significantly loosen this limitation by allowing large highly-deformed configurations, the theory still imposes practical limitations on pattern size and thickness. For example, extreme curvatures

Table 3. Simulation timings for large-scale simulations with timestep Δt in seconds, average seconds per step, and average seconds per frame for a reference framerate of 30fps.

Simulation		Δt	sec/step	sec/frame
satin small drape	Fig. 19	3.34e-04	1.10	109.72
stock. small drape	Fig. 19	3.34e-04	2.01	200.32
sweater basket	Fig. 18	1.67e-03	7.40	147.62
sweater honey	Fig. 18	1.67e-03	7.43	148.36
sweater rib	Fig. 18	8.35e-04	5.79	231.15
sweater satin	Fig. 18	1.67e-03	7.45	148.79
sweater stock.	Fig. 18	1.67e-03	7.44	148.50
shirt stock. small	Fig. 18	8.35e-04	3.74	149.19
shirt satin small	Fig. 18	1.67e-03	4.96	99.06
scarf	Fig. 1	8.35e-04	0.91	36.31
yarn bunny	Fig. 20	6.68e-04	1.23	61.20
yarnmadillo	Fig. 20	5.57e-04	11.16	668.24

at the macroscale may cause excessive self-intersections at the microscale. Similarly, approximating voluminous yarn patterns with a triangle-based cloth solver may make the garment look unrealistically thin.

Our fitting procedure based on regularized splines aims to strike a balance between generality and robustness. Although we present a number of heuristics to increase the quality of the fit for nonconvex data, we do not offer any provable performance guarantees, and the approach is tailored to our application domain. Our focus in this work was to find one approach that yields stable simulations and reproduces the essential qualitative features of yarn-level cloth. Due to various approximations in fitting, we do not expect a perfect quantitative match. It would be interesting to devise experiments similar to real-world devices for measuring cloth material response. We hope that these aspects can be improved further in future work.

Finally, our homogenization procedure is not limited to yarn-level cloth and could be useful for animating other complicated multi-physics materials like layered quilts, layered elastic materials [?], skin tissue, and layered deployable shells [?]. Outside of computer graphics, our technique may be applicable to the homogenization of composite materials, micro-structured shells, and finite-element simulations. This approach may also be helpful as an intermediate step in inverse problems like the design and fabrication of functional fibrous materials.

ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers and the members of the Visual Computing Group at IST Austria for their valuable feedback. We also thank the creators of the Berkeley Garment Library [?] for providing garment meshes, [?] and [?] for the armadillo and bunny meshes, the creators of libWetCloth [?] for their implementation of discrete elastic rod forces, and Tomáš Skřivan for inspiring discussions and help with Mathematica code generation.

This research was supported by the Scientific Service Units (SSU) of IST Austria through resources provided by Scientific

Computing. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 638176. Rahul Narain is supported by a Pankaj Gupta Young Faculty Fellowship and a gift from Adobe Inc.

A ELIMINATION OF PERIODIC VARIABLES

The constraints on periodic vertex positions (20) and edge twists (25) can be concatenated into one constraint

$$\boldsymbol{C}\boldsymbol{q} = \boldsymbol{d}. \tag{38}$$

We eliminate periodic variables by parametrizing the nullspace of (38). We found that simply using a QR factorization for this is not numerically robust and produced dense matrices. However, we can exploit that each periodically copied degree of freedom is used in exactly one (sub-)constraint. As such, we can write $\boldsymbol{C}$ as

$$\boldsymbol{C} = \begin{pmatrix}\boldsymbol{I} & \boldsymbol{A}\end{pmatrix}\boldsymbol{P}, \tag{39}$$

where $\boldsymbol{P}$ is a permutation matrix that permutes the columns of $\boldsymbol{C}$, splitting it into a left identity block $\boldsymbol{I}$ and a sparse right block $\boldsymbol{A}$; i.e., we bring the constraint matrix into reduced row echelon form, where the columns of $\boldsymbol{A}$ span the kernel of $\boldsymbol{C}$. In our case, the kernel represents exactly the original degrees of freedom that are copied. For our periodicity constraints, the matrix is of the form

$$\begin{pmatrix}\boldsymbol{I} & \boldsymbol{A}\end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_{3\times 3} & & -(\boldsymbol{R}^+)(\boldsymbol{R}^-)^\top & & \\ & 1 & & -1 & \\ & & \ddots & & \ddots \end{pmatrix}. \tag{40}$$

Note that $\boldsymbol{P}$ splits $\boldsymbol{q}$ into free variables $\boldsymbol{y}$ and copies $\hat{\boldsymbol{q}}_{\text{copy}}$:

$$\boldsymbol{P}\boldsymbol{q} = \begin{pmatrix}\hat{\boldsymbol{q}}_{\text{copy}} \\ \boldsymbol{y}\end{pmatrix}. \tag{41}$$

Then, we have

$$\boldsymbol{C}\boldsymbol{q} = \boldsymbol{d} \tag{42a}$$

$$\begin{pmatrix}\boldsymbol{I} & \boldsymbol{A}\end{pmatrix}\boldsymbol{P}\boldsymbol{q} = \boldsymbol{d} \tag{42b}$$

$$\begin{pmatrix}\boldsymbol{I} & \boldsymbol{A}\end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{q}}_{\text{copy}} \\ \boldsymbol{y}\end{pmatrix} = \boldsymbol{d} \tag{42c}$$

$$\hat{\boldsymbol{q}}_{\text{copy}} = -\boldsymbol{A}\boldsymbol{y} + \boldsymbol{d}, \tag{42d}$$

and

$$\boldsymbol{q} = \boldsymbol{P}^\top\begin{pmatrix}\hat{\boldsymbol{q}}_{\text{copy}} \\ \boldsymbol{y}\end{pmatrix} \tag{43a}$$

$$= \boldsymbol{P}^\top\begin{pmatrix}-\boldsymbol{A}\boldsymbol{y} + \boldsymbol{d} \\ \boldsymbol{y}\end{pmatrix} \tag{43b}$$

$$= \boldsymbol{P}^\top\begin{pmatrix}-\boldsymbol{A} \\ \boldsymbol{I}\end{pmatrix}\boldsymbol{y} + \boldsymbol{P}^\top\begin{pmatrix}\boldsymbol{d} \\ \boldsymbol{0}\end{pmatrix} \tag{43c}$$

$$= \quad \tilde{\boldsymbol{C}} \quad \boldsymbol{y} + \quad \tilde{\boldsymbol{d}}. \tag{43d}$$

Finally, we need to compute the initial $\boldsymbol{y}$ from $\boldsymbol{q}$. For generality, we consider the case when $\boldsymbol{C}_L\boldsymbol{q} \neq \boldsymbol{d}_L$ and find $\boldsymbol{y}$ as

$$\min_{\boldsymbol{y}} \left|\tilde{\boldsymbol{C}}\boldsymbol{y} + \tilde{\boldsymbol{d}} - \boldsymbol{q}\right|^2 \quad \text{s.t.} \quad \boldsymbol{C}_L(\tilde{\boldsymbol{C}}\boldsymbol{y} + \tilde{\boldsymbol{d}}) - \boldsymbol{d}_L = 0, \tag{44}$$

the solution of which is given by

$$\begin{pmatrix}\tilde{\boldsymbol{C}}^\top\tilde{\boldsymbol{C}} & \tilde{\boldsymbol{C}}^\top\boldsymbol{C}_L{}^\top \\ \boldsymbol{C}_L\tilde{\boldsymbol{C}} & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{y} \\ \boldsymbol{\lambda}\end{pmatrix} = \begin{pmatrix}\tilde{\boldsymbol{C}}^\top(\boldsymbol{q} - \tilde{\boldsymbol{d}}) \\ \boldsymbol{d}_L - \boldsymbol{C}_L\tilde{\boldsymbol{d}}\end{pmatrix}. \tag{45}$$